

mydbaccess Ver. 1.0.3 (May 2008) – introductory notes

mydbaccess is a GUI for MySQL databases featuring powerful and end-user-friendly search, query and reporting engine functionality. The application runs OUT-OF-THE-BOX after unpacking to a suitably PHP-enabled web-server publishing location.

The application is HTML, PHP and JavaScript'ed. It is **NOT** intended as database administration tool. It is designed to complement administration tools such as phpMyAdmin in focusing on **content based navigation, search features and reporting**. mydbaccess is coded around a standardized application framework based on database design and property enrichment.

This brief documentation is to introduce basic concepts of mydbaccess. **For a list of major improvements over previous versions please refer to the section 'release notes' at the end of this document.**

setting-up mydbaccess

mydbaccess is a server-side internet publishing application requiring MySQL and PHP processing capabilities. So you should have either a LAMP server, an appropriately configured Microsoft IIS or an other equivalent platform up and running. Make sure you have administration rights on your web and MySQL servers.

mydbaccess can be run from about any internet browser on a client system by calling mydbaccess.php at the publishing location. The client browser has to be javascript-enabled.

To install, just extract mydbaccess files to a folder within the internet publishing scope of your server and make sure the folder and files are publicly "read" accessible.

mydbaccess now runs "out of the box" as it will by default try to connect to the mydbaccess.net session administration service (**provided you have a live connection to the internet**).

IMPORTANT NOTE: running "OUT-OF-THE-BOX" is a testing feature and depends on mydbaccess connecting to the session manager at mydbaccess.net to establish any sever connections. This is intended to facilitate testing only! To avoid this, make sure you change the settings in the "myglobalsettings.inc" file in accordance with this manual once you've set up your own session manager database. Server access information such as USER NAMES and PASSWORDS submitted to the mydbaccess.net session manager will be stored only until session expiry. By using this form you agree to use mydbaccess.net services in accordance with the policies published on that site.

Please note, that the session administration service is presently being provided using an anonymous account for evaluation purposes only. The service may be terminated at any time without prior notification.

Please contact us on crm@mydbaccess.net if you wish to subscribe for an individual service account with mydbaccess.net.

If you do not have a subscription account you are strongly encouraged to set up your own session management database performing the following steps:

- have the MySQL command line client process the SQL statements in file **mysqlesessions.sql** by a user with database creation rights to set-up the session database;
- set-up a new user for session administration on your MySQL database;
- grant SELECT, INSERT and DELETE rights for this user on that database.

Connection and other settings may be adjusted to meet your requirements in the mydbaccess configuration file (myglobalsettings.inc).

mydbaccess.php may be called with a number of parameters, e.g. with login-information for the database you wish to use. You may use index.html (same as login.htm) or create your own webpage to submit such parameters. If called without parameters it will default to establish a connection to the sandbox database on mydbaccess.net (or any default database you may want to specify in the configuration file).

Login parameters accepted by mydbaccess.php (using either the POST or GET method) are:

Variable	Description
[dbuid]	User ID on MySQL database
[dbupwd]	Password to be used for authentication
[db]	Database to be accessed
[dbhost]	Host to the database you wish to access (e.g. mydbaccess.net:3306 for the mydbaccess sandbox)

Should you want to benefit from some of the more **powerful navigation and query engine features**, **make sure you also set-up mydbaccess system tables on your application database**: just collectively process the SQL statements in **systemtables.sql** on your target database to make the tables available.

System tables are addressed in more detail further down.

database design framework - Some key concepts

benefit from single-field primary-key organised databases

The efficiency of navigation features provided by the mydbaccess application is greatly enhanced by having your database tables organised adhering to the simple concept of:

- **referencing same information consistently using the same fieldname across tables** in your database,
- **with one of the tables declaring that particular field it's sole primary key.**

Starting from there, the following navigation principles apply:

1-n relationship navigation principle: Information stored to tables in a database is automatically associated through the consistent use of field names; i.e. mydbaccess navigates such data-set relationships based on fields referenced by the same name throughout the related sets.

Navigation automatically covers two steps:

- The navigation panel will display a selection of recordsets based on the current recordset (i.e. containing the field name of the primary key of the current recordset) and a selection of recordsets the active recordset depends upon (i.e. containing single field primary keys with field names occurring in the current recordset).
- Upon a jump to a related table, a content-based filter will automatically be applied to the target table limiting displayed records to the content of the related field in the active record of the calling recordset.

n-m relationship principle: Information in two distinct tables may be mutually associated in a many-to-many relationship. Such relationships are handled by noding tables, which are technically navigated just like 1:n links. Supporting node editing, **mydbaccess features automatic drop-down selectors based on primary key fields of the related tables.**

Recursive n-m relationship principle: Information stored to a table may be recursively **self-referenced** by registering such relationships using table "sysnm" (see system tables). As the linking table may further detail the linking relationship, the matching field names need to be specified.

mydbaccess automatically adds content-based combo-box referrals for any table registered for self-referencing lookups in table "sysnm".

design-based content processing

mydbaccess automatically processes display features derived from assumptions based on technical field properties and database relationships. Presently the following display features are derived from this concept: - Fields of BOOLEAN or TINYINT type are interpreted to be displayed as checkboxes. - Fields for which a table is found containing the field name as single primary key will be displayed as combo box only allowing content available in the related primary key table. - Fields of TEXT type and large VARCHAR fields are displayed as multi-line containers (TEXT field containers are expanded in function of stored information).

database element naming conventions

Special characters: mydbaccess is NOT designed to deal with blanks, dots, colons and similar stuff in any database, table or field descriptors.

Case-sensitivity: On case-sensitive systems the application will generally expect to deal with lower case table names.

Hiding system components: Databases, tables, querysets and applications are only included in drop-down combos available on the navigation panel, if the name of that database element does not begin with either "sys", "mysql" or "_" (underscore). In order to see such database elements, a user is required to have "GRANT" privileges on a database.

content display control and reporting engine: mydbaccess system tables

mydbaccess **does not depend on the availability of system tables** at all to function properly. However...

...system tables are required if you want to benefit from some of the more **powerful navigation and**

query engine features provided based on system tables as structured in **systemtables.sql**:

Framework Table	Description
sysfields	keeps standardised attribute information for field names used throughout a database
sysnm	keeps registration records of recursive n:m links on tables in a database
sysqueries	lets you store SQL statements
sysmakequerysets	groups queries stored to 'sysqueries' into a queryset allowing you to determine a processing sequence
sysquerysets	keeps references and descriptions of sets of queries defined by 'sysqueries' and 'sysmakequerysets'
sysapplications	keeps references to other webpages you wish mydbaccess to call from it's navigation panel
syshttpsessiondetails	just requires "ID" varchar(255) field to keep session ID; you may add any number of additional fields at your discretion to store and retrieve values of database-specific variables using mydbaccess php functions mySessionDetails_Set() and mySessionDetails_Get()

Setting up system tables is simple: just collectively process the SQL statements in **systemtables.sql** on your target database to make the tables available.

Working with system tables:

Determine, how database content is displayed: sysfields

The way how record contents is displayed in detail view is controlled in table "sysfields". Fields are assumed to consistently represent the atoms of information throughout a database. Accordingly, each field no matter in which and how many tables or recordsets it may feature, may be registered with additional features in table "sysfields" thus being treated consistently whenever a recordset containing this field is called. The following properties may presently be specified:

- Determine a **label** overriding the display of the real fieldname.
- Determine whether a field specifies a **link to a file or website**.
- Specify whether mydbaccess should try to display a linked file as **image** in detail view.
- Assign a **drop down look-up query** to a field (overrides default primary key based lookup).
- Determine, whether a field is **visible**.

Generating complex reports and database operation sequences: sysqueries, sysquerysets, sysmakequeryset:

mydbaccess lets you define sets of query sequences, i.e. SQL statements that are processed in a sequential order. **All such set definitions are automatically displayed and made available for processing on the mydbaccess navigation panel.**

To set up a set of queries - **even if consisting in a single query** - the following steps have to be performed:

- Create a new record in table **sysquerysets**, thus uniquely identifying and referencing your new

set of queries (at least specifying field “QuerySetID”).

- Create a record in table **sysqueries** for each of the queries required by the set (at least specifying fields “QueryID” and “SQL”, the latter containing a valid SQL statement).
- Finally, add records to **sysmakequeryset** linking your queries to the new set providing a processing order sequence.

database design framework: Image / File handling

mydbaccess uses the filesystem rather than BLOB fields to keep large chunks of information. Files may be uploaded and simply linked by filename reference to fields in database records. Filenames are presently assumed unique on a by table basis (this is likely to be further refined in future versions).

Fields to be assumed as filename-containers have to be registered once in the sysfields system table. PLEASE NOTE: If you want users to be able to upload and retrieve files, both folders 'userfiles' and 'sessions' should be r/w.

release notes

Current release status is **1.0.3 as from 22 May 2008**. For updates please refer to our project website at <http://mydbaccess.net>

The current release has included fixes for a few minor issues. In addition, the LICENSING SCHEME has been reviewed and adjusted to reflect GNU GPLv3 as the sole licensing mode.

In 1.0.2 some progress has been made over 1.0.1 as in particular:

- a deep-search functionality has been introduced enabling a user to search the entire database for a keyword in a single step
- slashes and quotes are now dealt with at insertion and search level
- and 'INSERT' & 'UPDATE' coding has been unified

Major fixes in 1.0.1 over 1.0.0 included:

- Fixed buggy InnoDB sort behavior in browser frame
- Navigation has been enhanced in that additional sessions (and browser windows) are opened in parallel when switching between related tables.
- Database controlled image / file handling is now table-based

disclaimer

The mydbaccess application, the underpinning database design framework, this documentation and related information is Copyright (c) 2006 – 2008 Felix Zickermann. The software comes licensed under the terms of the GNU General Public License, Ver. 3. You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

The application consists of all files contained in the compressed archive you downloaded from our project website on <http://sourceforge.net/projects/mydbaccess/>

If you have received this application from another source, it may differ from the original package.

Connections to our sample databases hosted by the mydbaccess.net website are not part of the application package and have been granted for convenience / trial purposes only.

Please refer to our terms & conditions governing the use of this software on our project website, which may be reviewed and amended by us from time to time. It is of your responsibility to keep checking for updates of our terms and conditions: Your continued use of our websites, software or services following the posting of changes to any of our policies, terms or conditions will mean you accept those changes as applicable to you.